

# iPhone + Drupal accessing your content within a native app

Peter Harrington

# Agenda

- Native apps or the mobile web? Where to invest my time?
- Configuring Drupal for web services
  - services and server modules
  - testing services
  - DEMO
- iPhone client coding
  - communicating with a remote web service
  - using a framework
  - DEMO
- Improved Security
- Speeding things up - improving the user experience
  - DEMO

# What we expect from an application

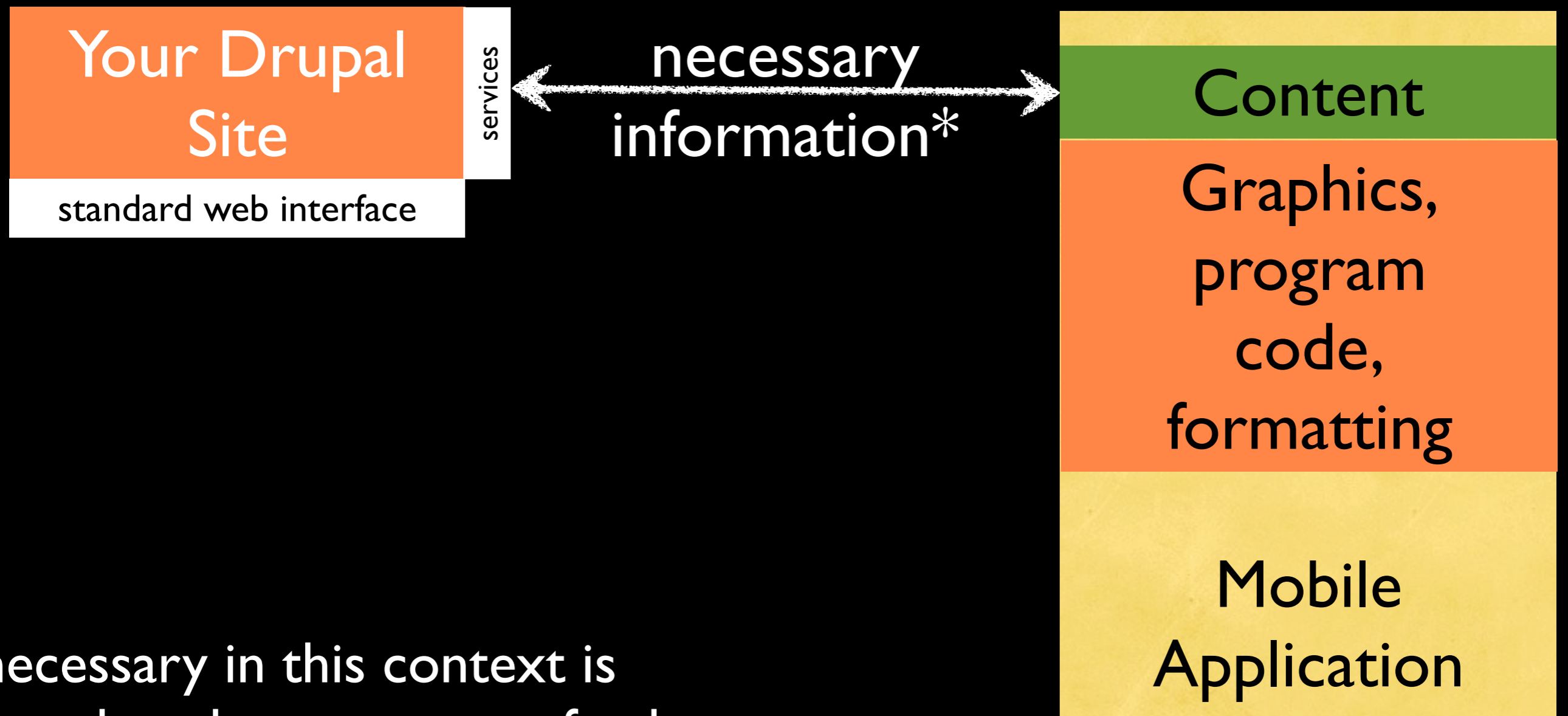
- storage space is unlimited
- fast
- realtime -- freshness
- a few commodities have shaped our expectations in an application
  1. non-volatile storage too cheap to measure
  2. bandwidth too cheap to measure



# mobile devices are not small PCs

- communication speeds are worse, you may have a 3G or 4G connection some of the time, but you may move in and out of covered areas
- memory is limited, and not the same type as a PC
- usage patterns are different
- the screen sizes are different

# How do we deliver what people EXPECT in a constrained environment?



\*necessary in this context is enough to keep your app fresh, yet responsive

# Drupal Services

- <http://drupal.org/project/services>
- comes with XMLRPC server, other servers such as JSON, REST, SOAP, and AMF can be downloaded from the above link.
- easy to setup
- no PHP scripting is needed to publish your own web service

# Installing Services

1. Download from <http://drupal.org/project/services>
2. move the folders into drupal/modules or drupal/sites/all/modules
3. on your Drupal site navigate to: admin/build/modules and enable Services, a server, and the specific services you want to use
4. go to admin/user/permissions and set permissions for the services you just enabled
5. navigate to admin/build/services and test the services

# Demo I

## locally testing services

# iPhone client side code

1. download the provided Xcode project:  
DrupalNiPhoneDemo
2. open `DruapNiPhoneDemoViewController.h`
3. replace the constants `DRUPAL_URL`, `USER_NAME`,  
and `PASSWD` with values relevant to your site.
4. build and run the program, and open the console to see the results

# iPhone client communication frameworks

- We are going to use a framework called XMLRPC, and it was originally written by Eric J. Czarny.
- This framework allows us to access remote procedures without the messy details of formatting, and communication protocols.

```
<value><struct>
<member><name>cid</name><value><string>19</string></value></member>
<member><name>pid</name><value><string>0</string></value></member>
<member><name>nid</name><value><string>2137</string></value></member>
<member><name>uid</name><value><string>0</string></value></member>
<member><name>subject</name><value><string>New Comment</string></value></member>
<member><name>comment</name><value><string>You are so funny. </string></value></member>
<member><name>hostname</name><value><string>71.202.128.83</string></value></member>
<member><name>timestamp</name><value><string>1261119735</string></value></member>
<member><name>status</name><value><string>0</string></value></member>
<member><name>format</name><value><string>0</string></value></member>
<member><name>thread</name><value><string>0a/</string></value></member>
<member><name>name</name><value><string></string></value></member>
<member><name>mail</name><value><string></string></value></member>
<member><name>homepage</name><value><string></string></value></member>
</struct></value>
```

```
cid = 19;
comment = "You are so funny. ";
format = 0;
hostname = "71.202.128.83";
nid = 2137;
pid = 0;
status = 0;
subject = "New Comment";
thread = "0a/";
timestamp = 1261119735;
uid = 0;
```

the framework sees this

and outputs this

# Demo II

## iPhone client code communicating with remote Drupal site

# Packaging Objects in XML

| iPhone side* | Drupal PHP |
|--------------|------------|
| NSArray      | Array      |
| NSDictionary | Object     |

```
//now do a node.save
NSMutableDictionary *nodeObj = [NSMutableDictionary dictionary];
[nodeObj setObject:@"location" forKey:@"type"];
[nodeObj setObject:poiTitle.text forKey:@"title"];
[nodeObj setObject:[NSString stringWithFormat:@"<p>%@</p>", poiNotes.text] forKey:@"body"];
[nodeObj setObject:USER_NAME forKey:@"name"];
//Next 5 lines are for proper packing of CCK field_long data
NSMutableDictionary *longitude = [NSMutableDictionary dictionary];
NSMutableDictionary *longitudeEl0 = [NSMutableDictionary dictionary];
[longitudeEl0 setObject:[NSString stringWithFormat:@"%f", location.longitude] forKey:@"value"];
[longitude setObject:longitudeEl0 forKey:@"0"];
[nodeObj setObject: longitude forKey:@"field_long"];
//repeat for field_lat
NSMutableDictionary *latitude = [NSMutableDictionary dictionary];
NSMutableDictionary *latitudeEl0 = [NSMutableDictionary dictionary];
[latitudeEl0 setObject:[NSString stringWithFormat:@"%f", location.latitude] forKey:@"value"];
[latitude setObject:latitudeEl0 forKey:@"0"];
[nodeObj setObject: latitude forKey:@"field_lat"];
//send it out
[reqUserInfo setMethod:@"node.save" withObjects:[NSArray arrayWithObjects:sessionID, nodeObj, nil]];
XMLRPCResponse *Response3 = [XMLRPCConnection sendSynchronousXMLRPCRequest:reqUserInfo];
NSString *source3 = [Response3 source];
NSLog(@"The node.save response is: %@", source3);
```

\*These relationships are specific to the XMLRPC framework used in the demo, and may be different for other frameworks.

# Improved Security

- encrypting outgoing user login credentials:

First enable keys under admin/build/services/settings on your Drupal site

Second do a 256 bit hash in the client code:

```
//For user.login we need: hash, domain_name,  
domain_time_stamp, nonce, sessid, username, passwd  
  
//generate hash sha256
```

See [3] for more details

- storing credentials may even leave you vulnerable to a binary dump

```
#define USER_NAME @"chdaBoy"  
#define PASSWD @"qwertyuiop"
```

# Accelerating the Communication with Drupal

- The default services are built for PC to PC communication. (Perhaps something more powerful.)
- Mobile devices don't have the hardware or bandwidth of a PC.

1 system.connect

2 user.login

3 your command

4 user.logout

each one of these is a client-server roundtrip



# Things we can do to speed up communications

1. increase the speed of light
2. tell our hosting provider to buy better equipment
3. tell our hosting provider to treat us as if we were the only customer
4. decrease the number of round trips
5. decrease the amount data coming and going
6. pack the data more densely

# Creating/Modifying Modules

- three files: `module_name.inc`, `module_name.module`, and `module_name.info`
- `module_name.info`: used during install
- `module_name.module`: sets up permissions, help text, tells which php function should be called, and what the inputs and output of each function should be.
- `module_name.inc`: the actual PHP code

# comment\_service.module

```
/**
 * Implementation of hook_help().
 */
function comment_service_help($path, $arg) {
  switch ($path) {
    case 'admin/help#services_comment':
      return t('<p>Provides comment methods to services applications. Requires services.module.</p>');
    case 'admin/modules#description':
      return t('Provides comment methods to services applications. Requires services.module.');
```

supplies simple help  
text

allows you to set  
permissions from  
Drupal/admin

gives specific  
details for each  
service

# comment\_service.inc

```
* Returns the comments of a specified node.
*
* @param $nid
* Unique identifier for the node.
* @param $count
* Number of comments to return.
* @param $start
* Which comment to start with. if present, $start and $count are used together
* to create a LIMIT clause for selecting comments. This could be used to do paging.
* @return
* An array of comment objects.
*/
function comment_service_load_node_comments($nid, $count = 0, $start = 0) {
    $comments = array();
    $limit = ((int)$count > 0 ? ' LIMIT '. (int)$start .', '. (int)$count .' ' : '');

    $result = db_query("SELECT cid FROM {comments} WHERE nid = %d ORDER BY thread DESC". $limit, $nid);
    while ($comment = db_fetch_array($result)) {
        $comments[] = _comment_load($comment['cid']);
    }

    return $comments;
}

/**
 * Returns the comments in terse form of a specified node.
 *
 * @param $nid
 * Unique identifier for the node.
 * @param $count
 * Number of comments to return.
 * @param $start
 * Which comment to start with. if present, $start and $count are used together
 * to create a LIMIT clause for selecting comments. This could be used to do paging.
 * @return
 * An array of comment objects, with the single value comment string.
 */
function comment_service_load_node_comments_terse($nid, $count = 0, $start = 0) {
    $comments = array();
    $limit = ((int)$count > 0 ? ' LIMIT '. (int)$start .', '. (int)$count .' ' : '');

    $result = db_query("SELECT cid FROM {comments} WHERE nid = %d ORDER BY thread DESC". $limit, $nid);
    while ($comment = db_fetch_array($result)) {
        $comments[] = db_fetch_object(db_query("SELECT comment, timestamp FROM {comments} WHERE cid = %d", $comment['cid']));
        //$comments[] = _comment_load($comment['cid']);
    }

    return $comments;
}
```

In addition to this code I set in comment\_service.module '#auth' => FALSE, which allows me to bypass the system.connect and user.login commands

# loadNodeComTerse total communications

```
[Session started at 2010-01-21 09:08:09 -0800.]
2010-01-21 09:08:10.981 DruaplNiPhoneDemo[1470:207] The buffer inside encode is: <?xml version="1.0"?
  ><methodCall><methodName>comment.loadNodeComTerse</methodName><params><param><value><i4>2137</i4></value></
  param><param><value><i4>10</i4></value></param><param><value><i4>0</i4></value></param></params></
  methodCall>
2010-01-21 09:08:12.446 DruaplNiPhoneDemo[1470:207] the comment.loadNodeComTerse response is <?xml
  version="1.0"?>

<methodResponse>
  <params>
    <param>
      <value><array><data>
        <value><struct>
          <member><name>comment</name><value><string>You are so funny. </string></value></member>
          <member><name>timestamp</name><value><string>1261119735</string></value></member>
        </struct></value>
        <value><struct>
          <member><name>comment</name><value><string>OMG! It's working!</string></value></member>
          <member><name>timestamp</name><value><string>1261119264</string></value></member>
        </struct></value>
        <value><struct>
          <member><name>comment</name><value><string>This is pretty nice. My boy!</string></value></member>
          <member><name>timestamp</name><value><string>1261119165</string></value></member>
        </struct></value>
        <value><struct>
          <member><name>comment</name><value><string>This is cool!.</string></value></member>
          <member><name>timestamp</name><value><string>1261117894</string></value></member>
        </struct></value>
      </data></array></value>
    </param>
  </params>
</methodResponse>
```

# DEMO III

## increased speed

# Improving the iPhone client side code

- run remote procedure calls in a separate thread

```
[NSThread detachNewThreadSelector:@selector(submitStuff) toTarget:self  
withObject:nil];
```

```
- (void)submitStuff{  
  
    NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];  
  
    //your communication with Drupal code goes here  
  
    [pool release];  
  
} //end submitStuff
```

# Improving the iPhone client side code (cont.)

- It's also a good idea to give the user some indication as to what is going on e.g. activity indicators

```
[progressIndicator startAnimating];
```

```
[progressIndicator stopAnimating];
```

It's usually bad practice to do update the UI in separate threads, so try to do these in the original thread.

- clean up memory leaks

```
[progressIndicator release];
```

# References

1. "Beginning iPhone Development" by Dave Mark and Jeff LaMarche Apress 2009
2. [http://www.morganstanley.com/institutional/techresearch/pdfs/2SETUP\\_12142009\\_R1.pdf](http://www.morganstanley.com/institutional/techresearch/pdfs/2SETUP_12142009_R1.pdf)
3. Overview of how to use SHA256 in Drupal: <http://fitc.sisutastic.com/utilizing-api-keys>
4. a simple way to test your JSON server from the command line: curl <http://drupal.org/node/305799>

# Native Apps Vs. the Mobile Web

| Native Apps  | Mobile Web  |
|--|---|
| any type of data you want to store is stored until the app is uninstalled* | data is cached until the device gets low on memory* |
| iterations take 10-14 days   | agile development                                   |

\*these are limitations from the hardware